Aplicații la formula lui Taylor

P1. Să se scrie formula lui MacLaurin pentru funcţia $f : [a, \infty] \to \mathbb{R},\ f(x) = \sqrt{a + x},\ a > 0.$

```
syms a x f expo xi
assume(a>0)
expo=sym(1)/sym(2);
f=a^expo*(1+x/a)^expo;
taylor(f,x,0,'Order',11)
```

ans =

$$\frac{x}{2\ \sqrt{a}} + \sqrt{a} - \frac{x^2}{8\,a^{3/2}} + \frac{x^3}{16\,a^{5/2}} - \frac{5\,x^4}{128\,a^{7/2}} + \frac{7\,x^5}{256\,a^{9/2}} - \frac{21\,x^6}{1024\,a^{11/2}} + \frac{33\,x^7}{2048\,a^{13/2}} - \frac{429\,x^8}{32768\,a^{15/2}} + \frac{715\,x^9}{65536\,a^{17/2}} \cdot$$

```
taylor(f,x,0,'Order',11)+subs(diff(f,x,11),x,xi)
```

ans =

$$\frac{654729075}{2048\,a^{21/2}\,\left(\frac{\xi}{a}+1\right)^{21/2}} + \frac{x}{2\ \sqrt{a}} + \sqrt{a} - \frac{x^2}{8\,a^{3/2}} + \frac{x^3}{16\,a^{5/2}} - \frac{5\,x^4}{128\,a^{7/2}} + \frac{7\,x^5}{256\,a^{9/2}} - \frac{21\,x^6}{1024\,a^{11/2}} + \frac{33\,x^7}{2048\,a^{13/2}} - \frac{\ }{3}$$

P2. Să se determine numărul natural n, astfel ca pentru $a = 0$ şi $f : \mathbb{R} \to \mathbb{R},\ f(x) = e^x\ (T_n f)$ să aproximeze f în $[-1, 1]$ cu trei zecimale exacte.

```
syms n rest(n)
rest(n)=3/factorial(n+1);
for k=3:10
    r=vpa(rest(k));
    disp([k,r])
    if abs(double(r))<1e-3, break; end
end
```

$(3\quad 0.125)$
$(4\quad 0.025)$
$(5\quad 0.0041666666666666666666666666666667)$
$(6\quad 0.0005952380952380952380952380952381)$

Deci, $n = 6$ este suficient

```
dezvT=taylor(exp(x),x,0,'Order',k+1)
```

dezvT =

$$\frac{x^6}{720} + \frac{x^5}{120} + \frac{x^4}{24} + \frac{x^3}{6} + \frac{x^2}{2} + x + 1$$

```
vpa(subs(dezvT,x,1.0),10)
```

ans = 2.718055556

```
exp(1)
```

```
ans = 2.7183
```

P3. Să se aproximeze $\sqrt[3]{999}$ cu 12 zecimale exacte.

```
dg=digits;
digits(13);
syms f(x) R(n) ex
ex=sym(1)/sym(3);
f(x)=10*(1-x)^(sym(1)/sym(3));
R(n)=vpa(nchoosek(ex,n)*10^(-3*(n+1)));
R(2)
```

ans = $-0.0000000001111111111111$

```
R(3)
```

ans = $6.172839506173\text{e-}14$

$n = 4$ este suficient

```
P=taylor(f,x,0,'Order',4)
```

P(x) =

$$-\frac{50\,x^3}{81} - \frac{10\,x^2}{9} - \frac{10\,x}{3} + 10$$

```
vpa(subs(P,x,0.001))
```

ans(x) = 9.996665554938

```
vpa(999^ex)
```

ans = 9.996665554938

```
digits(dg)
```

Conditionarea rădăcinilor ecuaţiilor polinomiale

Radu Trîmbiţaş

30 aprilie 2020

1 Senzitivitatea rădăcinilor polinomiale

Vom considera condiţionarea unei rădăcini ξ a unei ecuaţii polinomiale $p(x)$. Prin aceasta se înţelege influenţa asupra lui ξ a unei perturbaţii a coeficienţilor polinomului $p(x)$:

$$p_\varepsilon(x) = p(x) + \varepsilon g(x),$$

unde $g(x) \neq 0$ este un polinom arbitrar nenul.

Se poate arăta că dacă ξ este o rădăcină simplă a lui p, atunci pentru pentru valori absolute suficient de mici ale lui ε există o funcţie analitică $\xi(\varepsilon)$, cu $\xi(0) = \xi$, astfel încât $\xi(\varepsilon)$ să fie o rădăcină simplă a polinomului perturbat $p_\varepsilon(x)$:

$$p\left(\xi\left(\varepsilon\right)\right) + \varepsilon g\left(\xi(\varepsilon)\right) \equiv 0.$$

De aici, prin derivare în raport cu ε, se obţine pentru $k := \xi'(0)$ ecuaţia

$$kp'\left(\xi(0)\right) + g\left(\xi\left(0\right)\right) = 0,$$

deci

$$k := \frac{-g(\xi)}{p'(\xi)}.$$

Astfel, până la o aproximare de ordinul întâi (adică, eliminând termenii în ε cu puteri mai mari decât 1), se obţine prin dezvoltarea Taylor a lui $\xi(\varepsilon)$ relaţia

$$\xi\left(\varepsilon\right) := \xi - \varepsilon \frac{g\left(\xi\right)}{p'\left(\xi\right)}. \tag{1}$$

În cazul unei rădăcini multiple ξ, de ordin m, se poate arăta că $p(x) + \varepsilon g(x)$ are o rădăcină de forma

$$\xi\left(\varepsilon\right) = \xi + h\left(\varepsilon^{1/m}\right),$$

unde, pentru $|t|$ mic, $h(t)$ este o funcţie analitică cu $h(0) = 0$. Derivând de m ori în raport cu t şi observând că $p(\xi) = p'(\xi) = \cdots = p^{(m-1)}(\xi) = 0$, $p^{(m)}(\xi) \neq 0$, din

$$0 \equiv p_\varepsilon\left(\xi\left(\varepsilon\right)\right) = p\left(\xi + h(t)\right) + t^m g\left(\xi + h\left(t\right)\right), \qquad t^m = \varepsilon,$$

pentru $k = h'(0)$ ecuaţia

$$p^{(m)}(\xi)\, k^m + m! g(\xi) = 0,$$

de unde

$$k := \left[-\frac{m! g(\xi)}{p^{(m)}(\xi)}\right]^{1/m}.$$

Din nou, până la o aproximare de ordinul I

$$\xi(\varepsilon) = \xi + \varepsilon^{1/m} \left[-\frac{m! g(\xi)}{p^{(m)}(\xi)}\right]^{1/m}. \tag{2}$$

Pentru $m = 1$ această formulă se reduce la (1) pentru rădăcini simple.

Să presupunem că polinomul $p(x)$ este dat sub formă de listă de coeficienţi

$$p(x) = a_0 x^n + \cdots + a_n.$$

Pentru

$$g_i(x) = a_i x^{n-i}$$

polinomul $p_\varepsilon(x)$ se obţine prin înlocuirea lui a_i în $p(x)$ cu $a_i(1+\varepsilon)$. Formula (2) ne dă următoarea estimare a efectului asupra rădăcinii ξ a unei erori relative ε asupra lui a_i:

$$\xi(\varepsilon) - \xi = \varepsilon^{1/m} \left[-\frac{m! a_i \xi^{n-i}}{p^{(m)}(\xi)}\right]. \tag{3}$$

Astfel, este clar că în cazul rădăcinilor multiple, schimbările în valoarea rădăcinii $\xi(\varepsilon) - \xi$ sunt proporţionale cu $\varepsilon^{1/m}$, $m > 1$, pe când în cazul unei rădăcini simple ele sunt proporţionale doar cu ε: rădăcinile multiple sunt întotdeauna prost condiţionate. Rădăcinile simple pot fi şi ele prost condiţionate. Aceasta se întâmplă dacă factorul lui ε în (3),

$$k(i, \xi) := \left|\frac{a_i \xi^{n-i}}{p'(\xi)}\right|,$$

este mare comparativ cu ξ, ceea ce se poate întâmpla şi în cazul unor polinoame în aparenţă „nevătămătoare”.

Exemplul 1 (Wilkinson 1959) *Rădăcinile $\xi_k = k$, $k = 1, 2, \ldots, 20$, ale polinomului*

$$p(x) = (x-1)(x-2)\ldots(x-20) = \sum_{i=0}^{20} a_i x^{20-i}$$

sunt bine separate. Pentru $\xi_{20} = 20$, găsim $p'(20) = 19!$, şi înlocuind coeficientul $a_1 = -(1 + 2 + \cdot + 20) = -210$ cu $a_1(1+\varepsilon)$ cauzează o schimbare estimată de

$$\xi_{20}(\varepsilon) - \xi_{20} = \varepsilon \frac{210 \times 20^{19}}{19!} \approx \varepsilon \cdot 0.9 \times 10^{10}.$$

Cele mai drastice schimbări schimbări sunt cauzate în ξ_{16} *prin perturbarea lui* a_5. *Deoarece* $\xi_{16} = 16$ *şi* $a_5 \approx 10^{-10}$,

$$\xi_{16}(\varepsilon) - \xi_{16} = -\varepsilon a_5 \frac{16^5}{4!15!} \approx \varepsilon \cdot 3.7 \times 10^{14}.$$

Aceasta înseamnă că rădăcinile lui p sunt atât de prost condiţionate încât chiar şi calculul cu o aritmetică de 14 cifre nu va garanta nici o cifră corectă în ξ_{16}.

Exemplul 2 *Prin contrast, rădăcinile polinomului*

$$p(x) = \sum_{i=0}^{20} a_i x^{20-i} := \prod_{j=1}^{20} \left(x - 2^{-j}\right), \qquad \xi_j = 2^{-j},$$

deşi nu sunt bine separate şi se acumulează spre zero, sunt toate bine condiţionate. De exemplu, schimbând a_{20} *în* $a_{20}(1+\varepsilon)$ *cauzează o variaţie a lui* ξ_{20} *care până la o aproximare de ordinul întâi poate fi delimitată sub forma*

$$\left|\frac{\xi_{20}(\varepsilon) - \xi_{20}}{\xi_{20}}\right| = \left|\varepsilon \frac{1}{(2^{-1}-1)(2^{-2}-1)\dots(2^{-19}-1)}\right| \leq 4|\varepsilon|.$$

Mai general, se poate arăta că pentru toate rădăcinile ξ_j *şi toate schimbările* $a_i \to a_i(1+\varepsilon)$

$$\left|\frac{\xi_j(\varepsilon) - \xi_j}{\xi_j}\right| \leq 64|\varepsilon|.$$

Totuşi, rădăcinile sunt bine condiţionate numai în raport cu schimbări relative ale coeficienţilor a_i *şi nu pentru schimbări absolute mici. Dacă înlocuim* $a_{20} = 2^{-210}$ *cu* $\bar{a}_{20} = a_{20} + \Delta a_{20}$, $\Delta a_{20} = 2^{-48}\left(\approx 10^{-14}\right)$ *– acesta poate fi considerată o schimbare absolută mică – atunci polinomul modificat are rădăcinile* $\bar{\xi}_i$ *cu*

$$\bar{\xi}_1 \dots \bar{\xi}_{20} = \overline{a}_{20} = 2^{-120} + 2^{-48} = \left(2^{162}+1\right)\xi_1 \dots \xi_{20}.$$

Cu alte cuvinte, există cel puţin un indice r cu $\left|\overline{\xi}_r/\xi_r\right| \geq \left(2^{162}+1\right)^{1/20} > 2^8 = 256$.

Trebuie accentuat că formula (3) se referă doar la senzitivitatea rădăcinilor unui polinom

$$p(x) := \sum_{i=0}^{n} a_i x^{n-i}$$

în reprezentarea uzuală a coeficienţilor. Există şi alte moduri de reprezentare a polinoamelor – de exemplu, ca polinoame caracteristice ale matricelor tridiagonale prin elementele matricei. Efectul unei modificări a parametrilor asupra rădăcinilor într-o astfel de reprezentare pate diferi printr-un ordin de mărime de cel din formula (3) . Condiţionarea rădăcinilor se defineşte întotdeauna cu un tip particular de reprezentare în minte.

Exemplul 3 *Se ştie că pentru orice matrice tridiagonală*

$$J = \begin{bmatrix} \alpha_1 & \beta_2 & & & & \\ \beta_2 & \alpha_2 & \ddots & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \alpha_{19} & \beta_{20} \\ & & & & \beta_{20} & \alpha_{20} \end{bmatrix}$$

al cărei polinom caracteristic este $p(x) \equiv (x-1)(x-2)\cdots(x-20)$, *schimbări relative mici în* α_i *şi* β_i *cauzează schimbări relative mici ale rădăcinilor* $\xi_j = j$. *În raport cu această reprezentare, toate rădăcinile sunt bine condiţionate, deşi în raport cu reprezentarea uzuală prin coeficienţi sunt foarte prost condiţionate.*

Bibliografie

[1] J. Stoer, R. Bulirsch, Introduction to Numerical Analysis, 2nd edition, Springer, 1992

Polynomial spline interpolation problem

Virgilius-Aurelian Minuță

Babeș-Bolyai University of Cluj-Napoca
Faculty of Mathematics and Computer Science
Department of Mathematics

Cluj-Napoca

3^{rd} February 2021

Introduction

The polynomial interpolation problem

Let $f : I \to \mathbb{R}$ be a function, where $I = [a, b] \subset \mathbb{R}$.
Let Δ be a subdivision upon the interval $[a, b]$

$$\Delta : a = x_1 < x_2 < \ldots < x_{n-1} < x_n = b$$

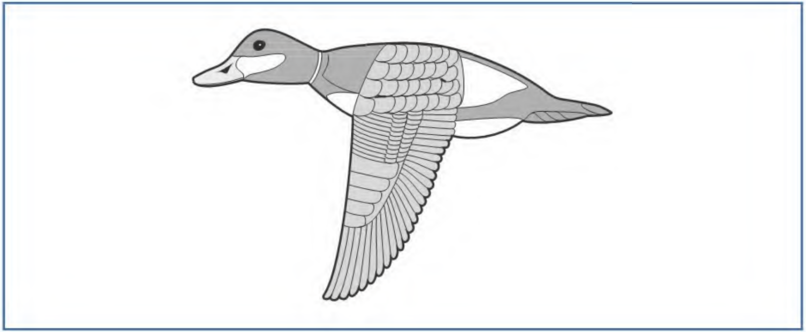
Given the values of f, $f_i = f(x_i)$, $i = \overline{1, n}$, find a function φ in a class of "approximations" Φ such that

$$\varphi(x_i) = f_i, \quad i = \overline{1, n}.$$

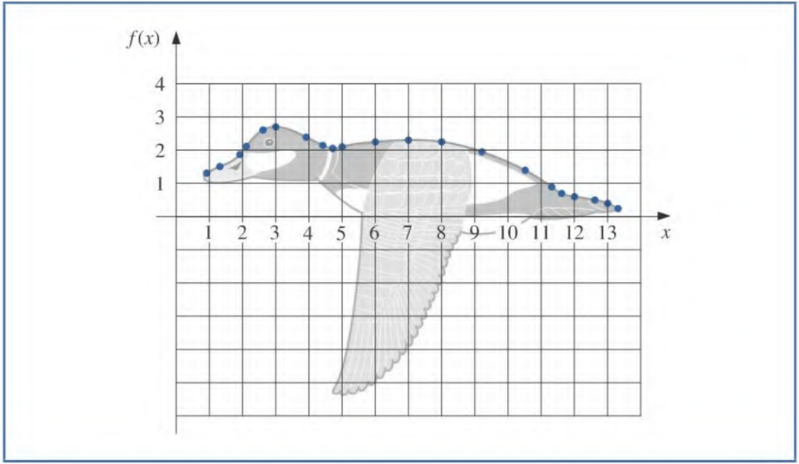
Example

If $\Phi = \mathbb{P}_m$ - the set of polynomials of degree at most m; we deal with polynomial interpolation. The interpolation problem is called here Lagrange interpolation and Hermite interpolation, respectively.

Application - Lagrange polynomial interpolation

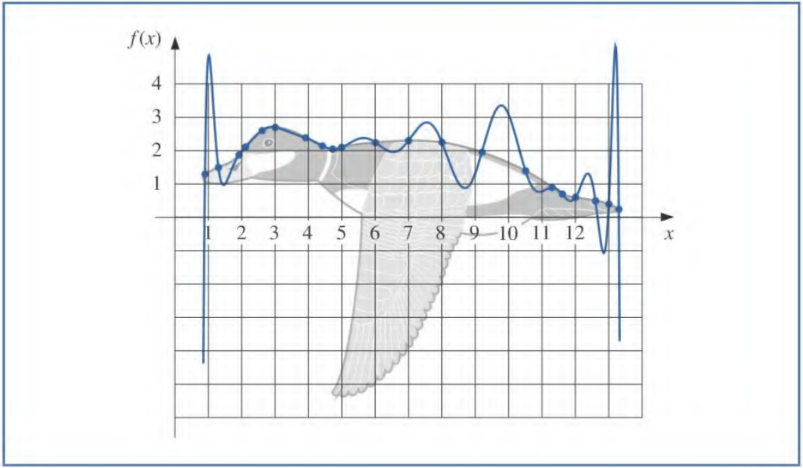


Application - Lagrange polynomial interpolation



x	0.9	1.3	1.9	2.1	2.6	3.0	3.9	4.4	4.7	5.0	6.0	7.0	8.0	9.2	10.5	11.3	11.6	12.0	12.6	13.0	13.3
$f(x)$	1.3	1.5	1.85	2.1	2.6	2.7	2.4	2.15	2.05	2.1	2.25	2.3	2.25	1.95	1.4	0.9	0.7	0.6	0.5	0.4	0.25

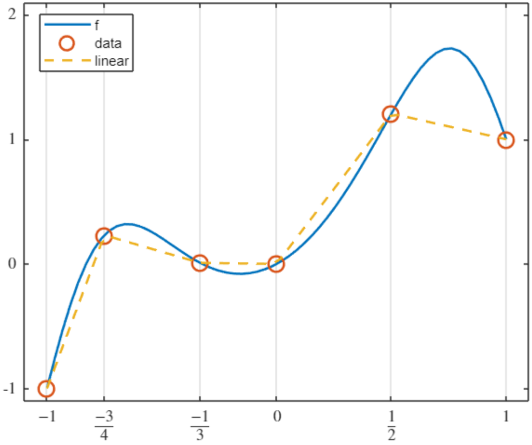
Application - Lagrange polynomial interpolation



Polynomial spline interpolation

Idea

- use low-degree polynomials on each subinterval $[x_i, x_{i+1}]$, $i = \overline{1, n-1}$



Polynomial spline interpolation

Polynomial spline functions

- Let
$$\Phi = \mathbb{S}_m^k(\Delta) := \{s \mid s \in C^k([a,b]),\ s|_{[x_i,x_{i+1}]} \in \mathbb{P}_m,\ i = \overline{1, n-1}\},$$
be the space of spline functions of degree $m \geq 0$ and smoothness class k relative to the subdivision Δ, where $k \in \mathbb{N} \cup \{-1\}$.
- We set $k = -1$ if we allow discontinuities at the joints.
- If $k = m$, then the functions $s \in \mathbb{S}_m^k(\Delta)$ are polynomials.

Linear spline interpolation

Linear spline interpolation

- For $m = 1$ and $k = 0$ we obtain linear splines.
- We want $s \in \mathbb{S}_1^0(\Delta)$ such that $s(x_i) = f_i$, $i = \overline{1, n}$.
- Solution: On the interval $[x_i, x_{i+1}]$

$$s(x) = f_i + (x - x_i) f[x_i, x_{i+1}]$$

- Interpolation error on the interval $[x_i, x_{i+1}]$:

$$|f(x) - s(x)| \leq \frac{(\Delta x_i)^2}{8} \max_{x \in [x_i, x_{i+1}]} |f''(x)|$$

- Interpolation error:

$$||f(x) - s(x)||_\infty \leq \frac{1}{8} |\Delta|^2 ||f''||_\infty$$

MATLAB Command

```
y=interp1(xi,fi,x,'linear')
```

Linear spline interpolation

Exercise 1

Write a MATLAB script that plots the graph of $f(x) = x + sin\pi x^2$ and computes the linear spline interpolant using the `interp1` command.

Solution

Linear spline interpolation

Solution

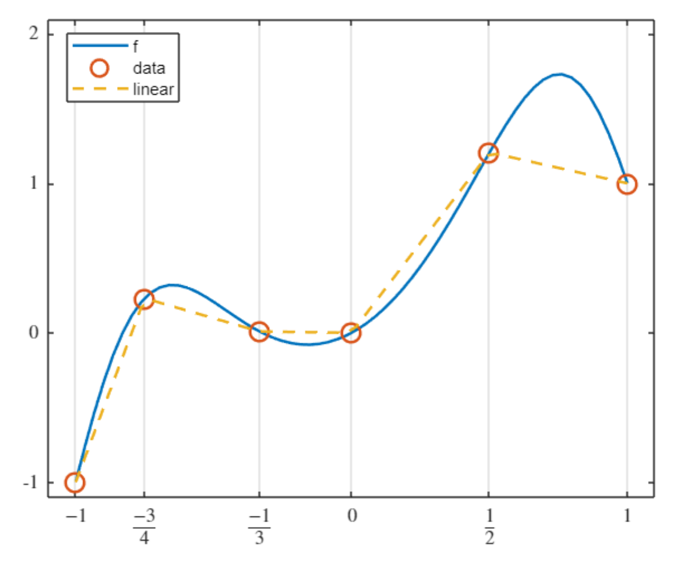
```
set(groot,'defaultLineLineWidth',1.5)
close all;
xi=[-1,-3/4,-1/3,0,1/2,1];
fi=f(xi);
x=linspace(-1,1,60);
y_f=f(x);
y_lin=interp1(xi,fi,x,'linear');
plot(x,y_f,'-',xi,fi,'o','MarkerSize',10);
hold on
plot(x,y_lin,'--')
xticks(xi)
yticks([-1,0,1,2]);
xticklabels({'$-1$','$\frac{-3}{4}$','$\frac{-1}{3}$','$0$','$\frac{1}{2}$','$1$'})
set(gca,'TickLabelInterpreter', 'latex')
set(gca,'XGrid','on')
axis([-1.1, 1.1, -1.1, 2.1])
legend('f','data','linear','Location','best')
hold off

```

```
function result = f(x)
result = x+sin(pi*x.^2);
end
```

Linear spline interpolation

Solution



Interpolation by cubic splines

Interpolation by cubic splines when $s \in \mathbb{S}_3^1(\Delta)$

- Continuity of the first derivative can be enforced by prescribing values for the first derivative at each point x_i.
- Let $m_1, m_2, \ldots, m_n$ be arbitrary given numbers.
- Denote $s(x)|_{[x_i,x_{i+1}]} = p_i(x), \quad i = \overline{1, n-1}$.
- We select p_i to be the (unique) solution of a Hermite interpolation problem:

$$\begin{array}{lll} p_i(x_i) = f_i & p_i(x_{i+1}) = f_{i+1} & i = \overline{1, n-1} \\ p_i'(x_i) = m_i & p_i'(x_{i+1}) = m_{i+1} & \end{array}$$

- Solution in Taylor form, for $x_i \le x \le x_{i+1}$:

$$p_i(x) = c_{i,0} + c_{i,1}(x - x_i) + c_{i,2}(x - x_i)^2 + c_{i,3}(x - x_i)^3$$

where

$$\begin{array}{lll lll} c_{i,0} & = & f_i & c_{i,1} & = & m_i \\ c_{i,2} & = & \frac{f[x_i,x_{i+1}] - m_i}{\Delta x_i} - c_{i,3}\Delta x_i & c_{i,3} & = & \frac{m_{i+1} + m_i - 2f[x_i,x_{i+1}]}{(\Delta x_i)^2} \end{array}$$

Interpolation by cubic splines

Piecewise cubic Hermite interpolation

- Here one sets $m_i = f'(x_i)$ (assuming that these derivative values are known).
- Interpolation error the interval $[x_i, x_{i+1}]$:

$$|f(x) - p_i(x)| \leq \left(\frac{1}{2}\Delta x_i\right)^4 \max_{x\in[x_i,x_{i+1}]} \frac{|f^{(4)}(x)|}{4!}$$

- Interpolation error:

$$||f(x) - s(x)||_\infty \leq \frac{1}{384}|\Delta|^4||f^{(4)}||_\infty$$

MATLAB Command

`y=interp1(xi,fi,x,'pchip')` - shape-preserving piecewise cubic interpolation $s \in \mathbb{S}_3^1(\Delta)$
We can also directly call the command `y=pchip(xi,fi,x)`

Interpolation by cubic splines

Exercise 2

Update the script from Exercise 1, to also compute a piecewise cubic interpolant using the `interp1` command.

Solution

Interpolation by cubic splines

Solution

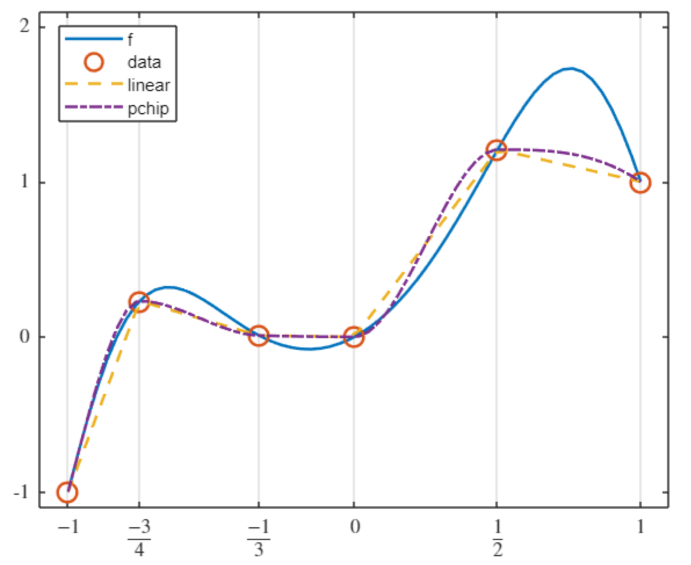
```
set(groot,'defaultLineLineWidth',1.5)
close all;
xi=[-1,-3/4,-1/3,0,1/2,1];
fi=f(xi);
x=linspace(-1,1,60);
y_f=f(x);
y_lin=interp1(xi,fi,x,'linear');
y_chip=interp1(xi,fi,x,'pchip');
plot(x,y_f,'-',xi,fi,'o','MarkerSize',10);
hold on
plot(x,y_lin,'--',x,y_chip,'-.')
xticks(xi)
yticks([-1,0,1,2]);
xticklabels({'$-1$','$\frac{-3}{4}$','$\frac{-1}{3}$','$0$','$\frac{1}{2}$','$1$'})
set(gca,'TickLabelInterpreter', 'latex')
set(gca,'XGrid','on')
axis([-1.1, 1.1, -1.1, 2.1])
legend('f','data','linear','pchip','Location','best')
hold off

```

```
function result = f(x)
result = x+sin(pi*x.^2);
end
```

Interpolation by cubic splines

Solution



Interpolation by cubic splines

Interpolation by cubic splines when $s \in \mathbb{S}_3^2(\Delta)$

- We have that

$$p''_{i-1}(x_i) = p''_i(x_i), \quad i \in \overline{2, n-1}$$

- Hence

$$2c_{i-1,2} + 6c_{i-1,3}\Delta x_{i-1} = 2c_{i,2}, \quad i \in \overline{2, n-1}$$

- We obtain the linear system (for $i \in \overline{2, n-1}$):

$$\Delta x_i m_{i-1} + 2(\Delta x_{i-1} + \Delta x_i)m_i + (\Delta x_{i-1})m_{i+1} = 3(\Delta x_i f[x_{i-1}, x_i] + \Delta x_{i-1} f[x_i, x_{i+1}])$$

- This is a system of $n-2$ linear equations with n unknowns $m_1, m2, \ldots, m_n$.
- By choosing m_1 and m_n, the system can be solved.

MATLAB Command

```
y=spline(xi,fi,x)
```

If `fi` has two more values than `xi` has entries, the first and last value play the role of m_1, m_n.

Interpolation by cubic splines when $s \in \mathbb{S}_3^2(\Delta)$

Complete (clamped) spline

- We take $m_1 = f'(a)$ and $m_n = f'(b)$
- Interpolation error, if $f \in C^4([a, b])$:

$$||f^{(r)}(x) - s^{(r)}(x)||_\infty \leq c_r|\Delta|^{4-r}||f^{(r)}||_\infty, \quad r = \overline{0, 3},$$

where $c_0 = \frac{5}{384}$, $c_1 = \frac{1}{24}$, $c_3 = \frac{3}{8}$ and c_3 is a constant depending on the ration $\frac{|\Delta|}{\max_i \Delta x_i}$.

Complete (clamped) spline

Exercise 3

Write a script that plots the graph of the $sin(x)$ function and compute a piecewise Hermitian interpolation together with a complete spline interpolation using the commands `pchip` and `spline`.

Solution

Complete (clamped) spline

Solution

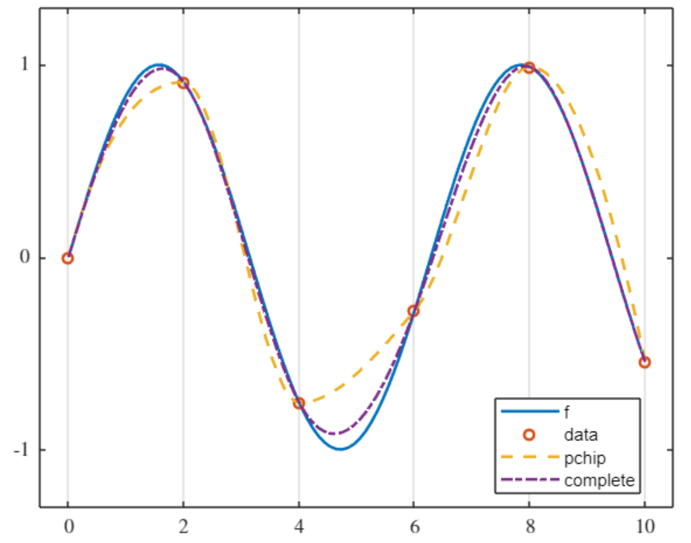
```
set(groot,'defaultLineLineWidth',1.5)
close all;
xi=0:2:10;
fi=f(xi);
x=linspace(0,10,100);
y_f=f(x);
y_pchip=pchip(xi,fi,x);
y_complete=spline(xi,[cos(0),fi,cos(10)],x);
plot(x,y_f,'-',xi,fi,'o','MarkerSize',5);
hold on
plot(x,y_pchip,'--',x,y_complete,'-.')
xticks(xi)
yticks([-1,0,1]);
set(gca,'TickLabelInterpreter', 'latex')
set(gca,'XGrid','on')
axis([-0.5, 10.5, -1.3, 1.3])
legend('f','data','pchip','complete','Location','best')
hold off

```

```
function result = f(x)
result = sin(x);
end
```

Complete (clamped) spline

Solution



Interpolation by cubic splines when $s \in \mathbb{S}_3^2(\Delta)$

Natural cubic spline

- We enforce $s''(a) = s''(b)$.
- We obtain 2 additional equations in the system

$$\begin{aligned} 2m_1 + m_2 &= 3f[x_1, x_2] \\ m_{n-1} + 2m_n &= 3f[x_{n-1}, x_n] \end{aligned}$$

- Advantage: No need to know derivative values.
- Disadvantage: Accuracy degradation at the endpoints (unless $f''(a) = f''(b) = 0$).

Interpolation by cubic splines when $s \in \mathbb{S}_3^2(\Delta)$

"Not-a-knot spline" (Carl de Boor)

- We enforce $p_1 \equiv p_2$ and $p_{n-2} \equiv p_{n-1}$
- This means that the first and last "knots" x_2 and x_{n-1} are inactive
- We again obtain 2 additional equations in the system:

$$\begin{aligned} \Delta x_2 m_1 + (\Delta x_2 + \Delta x_1) m_2 &= \gamma_1 \\ (\Delta x_{n-1} + \Delta x_{n-2}) m_{n-1} + \Delta x_{n-2} m_n &= \gamma_2, \end{aligned}$$

where

$$\begin{aligned} \gamma_1 &= \tfrac{1}{\Delta x_2 + \Delta x_1} (f[x_1, x_2] \Delta x_2 (3\Delta x_1 + 2\Delta x_2) + (\Delta x_1)^2 f[x_2, x_3]) \\ \gamma_2 &= \tfrac{1}{\Delta x_{n-1} + \Delta x_{n-2}} ((\Delta x_{n-1})^2 f[x_{n-2}, x_{n-1}] + (3\Delta x_{n-1} + 2\Delta x_{n-2}) \Delta x_{n-2} f[x_{n-1}, x_n]). \end{aligned}$$

MATLAB Command

`y=interp1(xi,fi,x,'spline')` or `y=spline(xi,fi,x)`

"Not-a-knot spline" (Carl de Boor)

Exercise 4

Update the script of Exercise 3, such that it plots a complete spline interpolation and a deBoor interpolation using the commands `interp1` and `spline`.

Solution

“Not-a-knot spline” (Carl de Boor)

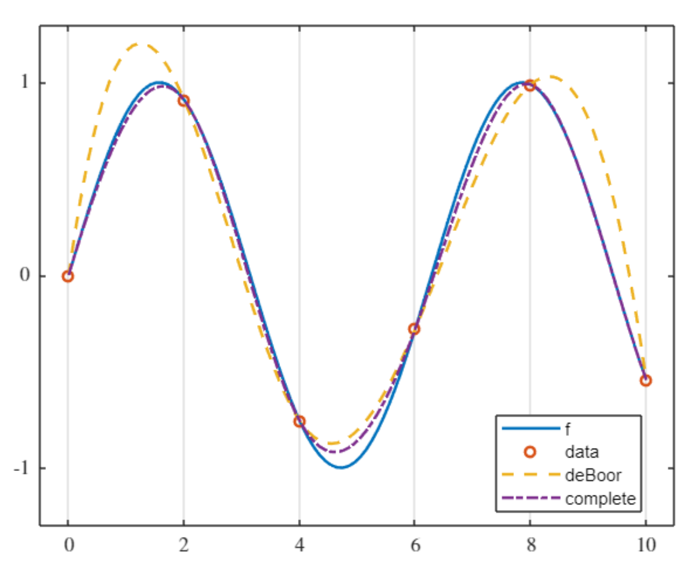
Solution

```
set(groot,'defaultLineLineWidth',1.5)
close all;
xi=0:2:10;
fi=f(xi);
x=linspace(0,10,100);
y_f=f(x);
y_deBoor=interp1(xi,fi,x,'spline');
y_complete=spline(xi,[cos(0),fi,cos(10)],x);
plot(x,y_f,'-',xi,fi,'o','MarkerSize',5);
hold on
plot(x,y_deBoor,'--',x,y_complete,'-.')
xticks(xi)
yticks([-1,0,1]);
set(gca,'TickLabelInterpreter', 'latex')
set(gca,'XGrid','on')
axis([-0.5, 10.5, -1.3, 1.3])
legend('f','data','deBoor','complete','Location','best')
hold off

function result = f(x)
result = sin(x);
end
```

"Not-a-knot spline" (Carl de Boor)

Solution



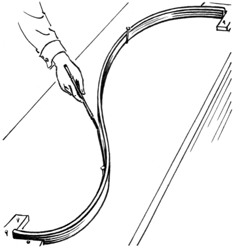
Interpolation by cubic splines when $s \in \mathbb{S}_3^2(\Delta)$

Theorem

For any function $g \in C^2([a,b])$ that interpolates f on Δ, we have

$$\int_a^b (g''(x))^2 dx \geq \int_a^b (s''_{\text{nat}}(x))^2 dx,$$

with equality if and only if $g = s_{\text{nat}}$.



References I

- R.T. Trîmbițas
  Numerical Analysis in MATLAB
  Presa Universitară Clujeană, Cluj-Napoca (2009)
- R.L. Burden, D.J. Faires, A.M. Burden
  Numerical Analysis
  Cengage Learning, Boston (2014)